**1. In your own words how would you characterize Wayne Babich’ “Shared Data” problem and what are the root causes of the problem? Why is it impossible to eliminate (even if you are only one person)?**

Shared data is a problem that occurs when multiple people are trying to work on the same material or data. One way to mitigate this (to an extent) is to use local branches of data. In our case, since we work via Teams and can see changes immediately, we should rarely run into an issue where 2 people are changing things concurrently without it being on purpose.

**2. In your own words how would you characterize Wayne Babich’ “Double Maintenance” problem and what are the causes of the problem? Why is it impossible to eliminate (even if you are only one person)?**

The double maintenance problem occurs when multiple people fix the same problem because they work on individual copies, or when one person has to fix the same problem on all copies. The way to mitigate this is to have ongoing merging on versions, which our cloud-system automatically does. We should also remember to communicate when we change something that impacts other team members.

**3. Discuss how the analysis of coordination problems from the lecture notes relates to your own personal experience from working in group(s).**

We have all experienced issues in which different people worked on the same problem and we ended up wasting a lot of time/work.

**4. Find places where the web-pages and the slides of this course “suffer” from Wayne Babich’ problems of “Shared Data” and “Double Maintenance”. Discuss different strategies for handling the problems and their advantages and drawbacks.**

**5. We created the concept of a workspace to protect ourselves from other people’s changes. But can the workspace always protect us from other people’s changes? Creating the workspaces, we get the “double maintenance” problem. Can we always handle that problem by merging?**

Not necessarily. Even if we have different workspaces and copies, if two team members are working on the same document, the two versions that they end up creating may be incompatible. When trying to merge the two, one member may have removed a feature that makes the other version work. We should always try to coordinate so that no two people are working on the exact same piece of data.

**6. We use the workspace to protect ourselves from other people’s changes, but how can we protect the repository from people’s bad changes?**

The repository is somewhat protected by default, since it keeps different versions of previous files. This means that we can always revert to a previous version for a file if it somehow rendered unusable. Apart from that, we have an integration manager who will check that everything is in an acceptable state upon being pushed to the main repository.

**7. List three situations from past experience where you have had problems that might have been caused by bad or missing SCM. Pick one and discuss the root causes for the problem and possible SCM solutions.**

Recently, one of our group members was working on a presentation as a teaching assistant for an exercise session. However, in the midst of setting up the presentation, another teaching assistant posted a finished version they had been making the same morning. Since the two TAs were not properly coordinated, they ended up with a double maintenance problem. Additionally, the slides were incompatible with one another, resulting in the team member’s work being unmergeable and thereby completely wasted.

**8. Discuss how the analysis of problems in handling changes from the lecture and lecture notes relates to your own personal experience from working in group (or alone).**

In every situation where a problem with SCM occurs, or in which we can imagine a significant SCM problem arising, it is important to take precautions in advance. By analysing problems and researching existing solutions we can agree on conventions, tools and methods that save us a lot of time and trouble.

**9. Brainstorm the following questions (in the context of a specific project you have been on): a. What are the goals regarding configuration management? b. What do you need to do to accomplish them (tools, money, time, people)? c. What do you need to do first? Next? If you are a manager? If you are a developer? d. How will you know when you have met your goals?**

a. The goal is to have a work environment in which we avoid impeding each other's work and where we keep up to date with the ongoing changes in the project. We also make sure that fatal errors cannot occur in which we would lose a lot of progress, and we keep different versions of files that are both easy to find and help us mitigate unwanted changes. We also assign responsibility to make sure that people are not working on the same data to mitigate wasted worktime and merge issues.

b. We need to spend time on the proper establishment of, and complying with, document convention, and on the delegation of tasks and responsibilities. We need dedicated people to take care of this job and to manage the process and our tools as we proceed. We also need the cloud services offered by teams and the configuration management and versioning tools of git.

c. Managers need to establish convention and decide upon the tools that will be used. Developers need to apply the established conventions to their work methods and familiarize themselves with the chosen tools to use them competently.

d. When we never have any processual issues. Most likely, we will be able to keep improving during the whole project.

**10. Discuss what kind of coordination problems there may occur between the different groups of people on a project – programmers, testers, designers, requirements engineers, QA, …. Discuss to what degree the concepts of baseline and traceability could be (partial) solutions to (some of these) problems – and how it should be handled.**

Many issues can occur. It is very possible that testers can end up testing an obsolete version of the software if updates are not pushed to them appropriately. It is also possible for members of our team to have a lot of downtime if their work can only proceed once other members have finished certain parts of the code, and they have different work structures or working days. Or one feature is deemed more important by one member than another, thus resulting in different prioritization and half-finished features. There is also the issue that work proceeds on a feature only for QA to find a fatal bug in a previous version of the code, rendering the new work unusable.